The Software Process

Software development is broken into phases (here we only consider object-oriented software development):

- Requirements Gathering
- Object-Oriented Analysis
- Object-Oriented Design
- Implementation
- Integration
- Maintenance
- Retirement

Notes:
- No testing phase – testing happens at each phase.
- No documentation phase – documentation is generated at each phase throughout the development.

Requirements Gathering

Assumption – the software being considered is economically justifiable – the job of the customer. During requirements gathering determine what the client *needs*, not what the client wants – concept exploration

Testing – rapid prototyping.

Documentation
- Rapid prototype
- Requirements document

Object-Oriented Analysis

Analysis
- Determine what the product is supposed to do – use cases, scenarios
- Extract high-level objects
- Extract formal specification of system behavior

Testing – requirements traceability, review

Documentation – high-level design document.
**Object-Oriented Design**
Analysis – what
Design – how

Many different aspects:
- Detailed class diagrams
- Sequence diagrams
- Collaborative diagrams

Testing – requirements traceability, review

Documentation – design documentation

**Implementation**
Implement detailed design in code

Testing – review, test cases (formal and informal)

Documentation – source code, test cases (with expected outcome)

**Integration**
Combine objects/modules and check product as a whole.

Testing
- Product testing
- Acceptance testing (special test cases stipulated by customer)

Documentation
- Commented source code
- Test cases for product as a whole

**Maintenance**
- Maintenance is any change to the software system once the client has accepted the software.
- Most money and effort is devoted to this phase
  - Most expensive phase to change code and fix problems

Testing – regression testing

Documentation
- Record of all changes made and why
- Regression test cases
**Retirement**

Good software is maintained,
But software becomes unmaintainable because:

- A drastic change in design has occurred
- The product has to be implemented on a totally new hardware platform/operating system
- Documentation is missing or inaccurate

Note: true retirement is rare.
The Cost of Fixing a Software Problem

There is an interesting relationship between development phase and fault repair cost.

Therefore, there is a real (as in monetary) motivation to find and fix as many issues/problems as early in the development process as possible.